IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants:	Jim Pruyne, et al.	Examiner: Asad M. Nawaz
Serial No.:	10/066,479	Group Art Unit: 2155
Filed:	January 30, 2002	Docket No.: 10006791-1
Title:	Method and System for Providing Exactly-Once Semantics for Web-Based Transaction Processing	

APPEAL BRIEF UNDER 37 C.F.R. § 41.37

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Appeal Brief is filed in response to the Notification of Non-Compliant Appeal Brief mailed October 5, 2006.

This Appeal Brief is also filed in response to the Final Office Action mailed October 25, 2005 and Notice of Panel Decision from Pre-Appeal Brief Review dated May 3, 2006.

AUTHORIZATION TO DEBIT ACCOUNT

It is believed that no extensions of time or fees are required, beyond those that may otherwise be provided for in documents accompanying this paper. However, in the event that additional extensions of time are necessary to allow consideration of this paper, such extensions are hereby petitioned under 37 C.F.R. § 1.136(a), and any fees required (including fees for net addition of claims) are hereby authorized to be charged to Hewlett-Packard Development Company's deposit account no. 08-2025.

I. REAL PARTY IN INTEREST

The real party-in-interest is the assignee, Hewlett-Packard Company, a Delaware corporation, having its principal place of business in Palo Alto, California.

II. RELATED APPEALS AND INTERFERENCES

There are no known related appeals or interferences known to appellant, the appellant's legal representative, or assignee that will directly affect or be directly affected by or have a bearing on the Appeal Board's decision in the pending appeal.

III. STATUS OF CLAIMS

Claims 1 – 23 are canceled, and claims 24 – 45 stand finally rejected. The rejection of claims 24 – 45 is appealed.

IV. STATUS OF AMENDMENTS

No amendments were made after receipt of the Final Office Action. All amendments have been entered.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The following provides a concise explanation of the subject matter defined in each of the claims involved in the appeal, referring to the specification by page and line number and to the drawings by reference characters, as required by 37 C.F.R. § 41.37(c)(1)(v). Each element of the claims is identified by a corresponding reference to the specification and drawings where applicable. Note that the citation to passages in the specification and drawings for each claim element does not imply that the limitations from the specification and drawings should be read into the corresponding claim element or that these are the sole sources in the specification supporting the claim features.

Claim 24

Claim 24 is directed to a method for performing a web transaction. Fig. 2 illustrates an interaction protocol for processing web-transactions. A first element of the claim recites "obtaining a form that includes a unique identifier for the web transaction"

(FIG. 2, Step 1: p. 13, lines 1-21). As shown in Fig. 2, the client requests a web page that contains a form. A second element of the claim recites "initiating a database update and generating a log for the database update such that the log is identified by the unique identifier" (FIG. 2, Step 2: p. 13, lines 15-21; p. 18, lines 6-19). As shown in Fig. 2, a web server send a client a form embedded with a UUID. The user completes the form and sends it and the UUID back to the server. The server returns the UUID, data, and timestamp to the client. A third element of the claim recites "obtaining a request to reload a status page such that the request includes the unique identifier" (FIG. 2, Step 3: p. 14, lines 1-25). The status page informs the user of the status of the transaction. The status page can include the UUID, the data entered into the form, and a timestamp. The status page is set to automatically reload after a predetermined time interval. The user may also manually reload the page by issuing a page refresh command. A fourth element of the claim recites "accessing the log in response to the request and retrying the database update if the log indicates a failure of the database update such that the database update is performed at most once" (p. 16, lines 3-19). This element describes what happens if a failure occurs at step 2 in Fig. 2.

Claim 25

Claim 25 is directed to the method of claim 24 and further defines obtaining a form. The claim recites two elements that include "obtaining the form in a post command from a client" (p. 13, lines 1-3), and "providing the status page to the client in response to the post command such that the status page includes the unique identifier" (p. 13, lines 6-27). As noted, a client requests a web page that contains a form.

Claim 26

Claim 26 is directed to the method of claim 25, wherein the request to reload is automatically generated by the status page at the client (p. 14, lines 3-10 and 23-25).

Claim 28

Claim 28 is directed to the method of claim 25, wherein the request to reload includes a set of data for retrying the database update (p. 14, lines 15-19).

Claim 30

Claim 30 is directed to the method of claim 24, wherein retrying the database update includes rolling back the database update after a timeout period and then retrying the database update (p. 14, line 26 – p. 16, line 16).

Claim 31

Claim 31 is directed to the method of claim 30. Claim 31 further comprises the element of determining the timeout period in response to a timestamp contained in the status page (p. 14, lines 20-22). The status page can also include a timestamp that the server uses to determine if the predetermined timeout period elapsed.

Claim 32

Claim 32 is directed to a web transaction processing system. This system is shown in Fig. 1 (see FIG. 1, #100). The system includes two claim elements: a database and a server. The database (FIG. 1, #130) holds a set of data associated with a web transaction (p. 7, line 27 – p. 8, line 6). For example, in order to execute browser requests, the web server (FIG. 1, #124) runs transactions against the database (#130). The server (FIG. 1, #124) performs several functions: The server obtains a form (FIG. 3, #374) that includes a unique identifier for the web transaction, initiates an update to the database in response to the form, and generates a log for the update such that the log is identified by the unique identifier. These functions are more fully explained in connection with Fig. 2. The server also has a retry mechanism (FIG. 1, #148). This retry mechanism obtains a request to reload a status page (FIG. 3, #320) such that the request includes the unique identifier and accesses the log in response to the request. The retry mechanism also retries the update if the log indicates a failure of the update such that the database update is performed at most once (p. 10, lines 13-26; p. 13, lines 12-21; p. 14, lines 1-25; p. 16, lines 3-19). As discussed more fully on page 16 of the specification, there are two possible cases to consider when the browser uses a web server that fails. In the first case, the browser receives the status page, and in the second case, the browser does not receive the status page. When the browser receives the status page, the reload logic performs the check

against the web server. When the browser does not receive this page, the browser times out and failure recovery occurs.

Claim 33

Claim 33 is directed to the web transaction processing system of claim 32. Claim 33 recites that the server obtains the form in a post command from a client and then provides the status page to the client such that the status page includes the unique identifier (p. 13, lines 6-27).

Claim 34

Claim 34 is directed to the web transaction processing system of claim 33. Claim 34 recites that the request to reload is automatically generated by the status page at the client (p. 14, lines 3-10 and 23-25).

Claim 36

Claim 36 is directed to the web transaction processing system of claim 33. Claim 36 recites that the request to reload includes the data for retrying the update (p. 14, lines 15-19).

Claim 37

Claim 37 is directed to the web transaction processing system of claim 32. Claim 37 recites that the retry mechanism rolls back the update after a timeout period and then retries the update (p. 14, line 26 – p. 16, line 16).

Claim 38

Claim 38 is directed to the web transaction processing system of claim 37. Claim 38 recites that the retry mechanism determines the timeout period in response to a timestamp contained in the status page (p. 14, lines 20-22).

Claim 39

Claim 39 is directed to a web transaction system. This system is shown in Fig. 1 (see FIG. 1, #100). This system includes two claim elements: a client and a server. The client (FIG. 1, #110) enters a set of data pertaining to a web transaction into a form (FIG. 3, #374) such that the form includes a unique identifier (FIG. 3, #378) for the web transaction (p. 13, lines 1-21). For instance, the client requests a form from the server. After receiving the form, the client enters data into the form and sends the completed form and UUID to the server. The server (FIG. 1, #124) obtains the form from the client and initiates an update to a database in response to the form. The server then generates a log for the update such that the log is identified by the unique identifier. In one embodiment, the server receives the form, executes biz-logic, and sends a status page to the browser. The server also has a retry mechanism (FIG. 1, #148). This mechanism obtains a request from the client to reload a status page (FIG. 3, #320) such that the request includes the unique identifier. The retry mechanism also accesses the log in response to the request and retries the update if the log indicates a failure of the update such that the database update is performed at most once (p. 10, lines 13-26; p. 13, lines 12-21; p. 14, lines 1-25; p. 16, lines 3-19). As discussed more fully on page 16 of the specification, there are two possible cases to consider when the browser uses a web server that fails. In the first case, the browser receives the status page, and in the second case, the browser does not receive the status page. When the browser receives the status page, the reload logic performs the check against the web server. When the browser does not receive this page, the browser times out and failure recovery occurs.

Claim 40

Claim 40 is directed to the web transaction system of claim 39. Claim 40 recites that the server obtains the form in a post command from a client and then provides the status page to the client such that the status page includes the unique identifier (p. 13, lines 6-27).

Claim 41

Claim 41 is directed to the web transaction system of claim 40. Claim 41 recites that the request to reload is automatically generated by the status page at the client (p. 14, lines 3-10 and 23-25).

Claim 43

Claim 43 is directed to the web transaction system of claim 40. Claim 43 recites that the request to reload includes the data for retrying the update (p. 14, lines 15-19).

Claim 44

Claim 44 is directed to the web transaction system of claim 40. Claim 44 recites that the retry mechanism rolls back the update after a timeout period and then retries the update (p. 14, line 26 – p. 16, line 16).

Claim 45

Claim 45 is directed to the web transaction system of claim 44. Claim 45 recites that the retry mechanism determines the timeout period in response to a timestamp contained in the status page (p. 14, lines 20-22).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 24 – 45 are rejected under 35 USC § 102(e) as being anticipated by USPN 6,259,701 (Shur).

VII. ARGUMENT

The rejection of claims 24 – 45 is improper, and Applicants respectfully requests withdraw of this rejection.

The claims do not stand or fall together. Instead, Applicants present separate arguments for various independent and dependent claims. Each of these arguments is separately argued below and presented with separate headings and sub-heading as required by 37 C.F.R. § 41.37(c)(1)(vii) as follows:

Group I: claims 24-45, with claim 24 representing the group;
Group II: claims 25, 33, and 40, with claim 25 representing the group;
Group III: claims 26, 34, and 41, with claim 26 representing the group;
Group IV: claims 28, 36, and 43, with claim 28 representing the group;
Group V: claims 30, 37, and 44, with claim 30 representing the group; and
Group VI: claims 31, 38, and 45, with claim 31 representing the group.

Claim Rejections: 35 USC § 102(e)

Claims 24 – 45 are rejected under 35 USC § 102(e) as being anticipated by USPN 6,259,701 (Shur). A proper rejection of a claim under 35 U.S.C. §102 requires that a single prior art reference disclose each element of the claim. See MPEP § 2131, also, *W.L. Gore & Assoc., Inc. v. Garlock, Inc.*, 721 F.2d 1540, 220 U.S.P.Q. 303, 313 (Fed. Cir. 1983). Since Shur does not teach each element in the claims, these claims are allowable over Shur.

As a precursor to the arguments, Applicants provide an overview of Shur and the independent claims.

Overview of Shur

As discussed in the background of Shur, unicast networks use point-to-point communication, and multicast networks use point-to-multipoint communication. Generally, clients in unicast networks cannot access communications in multicast networks. Shur solves this problem so unicast clients can participate in multicast session

communications (2: 64-67). Generally, Shur provides servers in the unicast network that function as gateways to the multicast network so unicast clients can access multicast sessions (3: 33-36).

Overview of Independent Claims

In performing a web transaction, a client receives a form from a server. The form includes a unique identifier that identifies the web transaction. The client enters data into the form and sends the form, the data, and the unique identifier to the server. Upon receiving the information from the client, the server then sends the client a status page and the unique identifier for the web transaction. The server also updates a database and generates a log for the update. The database update is retried if the log indicates a failure of the database update such that the database update is performed at most once.

Lack of Specificity in Office Action

Independent claim 24 recites numerous different elements, such as a form, a unique identifier, a log, and a status page. In rejecting claim 24, the Office Action merely cites a large section of Shur (namely, column 5 lines 21-65). The Office Action, however, never specifies which portions of Shur correspond to which elements in claim 24. In other words, the Office Action never argues or specifies which elements of Shur's system correspond with the words in claim 24, such as form, unique identifier, log, status page, etc. Applicants believe that such specificity is not stated because Shur does not teach a system or method that has elements as recited in claim 24. Additionally, Applicants admit that it is extremely difficult to rebut the Office Action since the action makes broad generalizations to reject the claims, instead of specifying which portions of Shur's teachings correspond to which elements in the claims. Nonetheless, Applicants will show that Shur does not teach the elements as recited in the claims.

Group I: claims 24-45

Independent claim 24 recites numerous recitations that are not taught in Shur. Applicants provide examples below and argue that success on any one of these examples warrants withdrawal of the rejection.

Example 1

Claim 24 recites "obtaining a form that includes a unique identifier for the web transaction." In other words, this recitation requires that **a form** is obtained. Shur does not teach this recitation. The Office Action cites Shur at col. 4, lines 41-64. Applicants respectfully disagree.

At column 4, lines 41-64, Shur teaches steps for a Unicast client to join a Multicast session. The client sends a server a request "for a copy of the page containing the Multicast sessions" (4: 50-52). The server sends a message to the client asking for a login ID and password (4: 52-54). If the client is authenticated, the server sends the client a list of Multicast sessions (4: 54-56).

Notice that the client in Shur never obtains "a form" as recited in claim 24 ("obtaining a form…"). In other words, the client in Shur receives a list of Multicast sessions. A list of sessions is not a "form."

For at least these reasons, the independent claims and their dependent claims are allowable over Shur.

Example 2

Claim 24 recites obtaining a form that "includes a unique identifier for the web transaction." In other words, this recitation requires that the **form includes a unique identifier for a web transaction**. Shur does not teach this recitation. The Office Action cites Shur at col. 4, lines 41-64. Applicants respectfully disagree.

At column 4, lines 41-64, Shur teaches steps for a Unicast client to join a Multicast session. The client sends a server a request "for a copy of the page containing the Multicast sessions" (4: 50-52). The server sends a message to the client asking for a login ID and password (4: 52-54). If the client is authenticated, the server sends the client a list of Multicast sessions (4: 54-56).

In Shur, the login ID and password are separate from the list of Multicast sessions. In other words, Shur never teaches that the list of Multicast sessions includes the login ID and password. Instead, once the client is authenticated, the client is sent the list. Shur does not even suggest that this list includes the login ID and password or any

unique identifier required for the web transaction. Again, claim 24 recites that the form "includes a unique identifier for the web transaction."

For at least these reasons, the independent claims and their dependent claims are allowable over Shur.

Example 3

Independent 24 recites "obtaining a request to reload a status page." Shur does not teach **reloading a status page**. The Office Action cites Shur at col. 5, lines 21-65. Applicants respectfully disagree.

This section of Shur teaches that the client completes a form and sends the completed form to the server (5: 57-59).[1] If the fields in the form are not correctly completed, then "an error message is returned to the client" (5: 60-61). Thus, Shur sends a client an error message. Sending a client an error message does not teach "reloading a status page." First, an error message is not a status page. Second, the error message is not "reloaded." In other words, the specification of Shur simply does not teach the recitations of the claim 24.

For at least these reasons, the independent claims and their dependent claims are allowable over Shur.

Example 4

Independent claim 24 recites obtaining a request to reload a status page "such that the request includes the unique identifier." Shur does not teach a request to reload a status page such that the **request itself includes the unique identifier for the web transaction**. As noted in connection with example 3, Shur sends a client an error if the form is not correctly completed. Nowhere whatsoever does Shur teach or even suggest that the error message includes the login ID of the client.[2]

For at least these reasons, the independent claims and their dependent claims are allowable over Shur.

---

[1] The Office Action never cites this section of Shur for teaching the first element of claim 24 ("obtaining a form that includes a unique identifier for the web transaction"). Regardless, the form itself in this cited section of Shur does not include a unique identifier for a web transaction.

[2] The Office Action equates the claimed "unique identifier" with Shur's login ID that the client sends to the server to create a new multicast session.

Examples 5-7

Independent claim 24 recites the following recitations:

> accessing the log in response to the request and retrying the database update if the log indicates a failure of the database update such that the database update is performed at most once.

Shur does not teach any of these recitations. The Office Action cites column 5, lines 21-65, but Applicants respectfully disagree for numerous reasons.

First, the recitation states "accessing the log in response to the request" to reload a status page. The Office Action has not cited a portion of Shur that corresponds to this recitation. Shur does teach that if the data in the form is correctly completed, then the server stores the data in a file that is indexed to the login id (5: 60-63). Shur, however, never teaches a request to reload a status page. Shur never teaches accessing the file (i.e., log) in response to a request to reload a status page.

For at least these reasons, the independent claims and their dependent claims are allowable over Shur.

Second, the recitation states "retrying the database update if **the log indicates a failure** of the database update" (emphasis added). The Office Action has not cited a portion of Shur that corresponds to this recitation. Shur does teach that if the data in the form is not correctly completed, then the server sends the client an error message (5: 60-61). Shur, however, never teaches retrying an update if the log itself indicates a failure. The log in Shur never indicates a failure.

For at least these reasons, the independent claims and their dependent claims are allowable over Shur.

Third, the recitation states that if the log indicates a failure, "the database update is **performed at most once**" (emphasis added). The Office Action has not cited a portion of Shur that corresponds to this recitation. Shur does teach that if the data in the form is correctly completed, then the server stores the data in a file that is indexed to the login id;

otherwise an error is sent (5: 60-63). Shur never teaches whatsoever that if the log indicates a failure, then the database update is performed at most once.

For at least these reasons, the independent claims and their dependent claims are allowable over Shur.

Group II: claims 25, 33, and 40

Dependent claim 25 recites that the form is obtained "in a post command from the client." In Shur, the client does sent a completed form back to the server. Shur, however, is completely silent about sending the form as a "post command."

The Office Action cites Shur at col. 5, line 57 – col. 6, line 3. Applicants respectfully ask the Board of Appeals to read this section of Shur. Where does Shur discuss obtaining a form in a "post command" from the client?

Group III: claims 26, 34, and 41

Dependent claim 26 recites that the request to reload is automatically generated **by the status page** at the client. Shur teaches that a client sends the form to a server. If the form is not completed correctly, then the server sends an error message to the client. Even assuming *arguendo* that Shur's error message is a status page (which it is not), nowhere does Shur teach that the error message automatically generates a request to reload at the client.

Group IV: claims 28, 36, and 43

Dependent claim 28 recites that the request to reload includes a set of data for retrying the database update. The Office Action argues that Shur teaches this recitation since Shur mentions numerous fields in the form and includes session ids, etc. Applicants respectfully argue that the Examiner is not considering all of the recitations as they actually appear in the claim. The claim recites that the **request to reload** itself includes a set of data for retrying the database update. In Shur, the numerous fields and session ids are not included in a request to reload.

Group V: claims 30, 37, and 44

Dependent claim 30 recites that retrying the database update includes rolling back the database update after a timeout period and then retrying the database update. The Office Action cites column 6, lines 15-41 in Shur. Nowhere does this section in Shur teach "rolling back the database update" or performing a roll back after a "timeout period."

Group VI: claims 31, 38, and 45

Dependent claim 31 recites determining a timeout period in response to a **timestamp** contained in the status page. The Office action cites column 5, lines 49-57. This section of Shur does teach that the form includes "a filed to set the Multicast packet Time to Live value." But this teaching has nothing to do with the claimed recitation "determining a timeout period in response to a **timestamp** contained in the status page."

CONCLUSION

In view of the above, Applicants respectfully request the Board of Appeals to reverse the Examiner's rejection of all pending claims.

Any inquiry regarding this Amendment and Response should be directed to Philip S. Lyren at Telephone No. 832-236-5529. In addition, all correspondence should continue to be directed to the following address:

Hewlett-Packard Company
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

Respectfully submitted,

/Philip S. Lyren #40,709/

Philip S. Lyren
Reg. No. 40,709
Ph: 832-236-5529

VIII. Claims Appendix

1-23 (Cancelled).

24. A method for performing a web transaction, comprising:

obtaining a form that includes a unique identifier for the web transaction;

initiating a database update and generating a log for the database update such that the log is identified by the unique identifier;

obtaining a request to reload a status page such that the request includes the unique identifier;

accessing the log in response to the request and retrying the database update if the log indicates a failure of the database update such that the database update is performed at most once.

25. The method of claim 24, wherein obtaining a form comprises:

obtaining the form in a post command from a client;

providing the status page to the client in response to the post command such that the status page includes the unique identifier.

26. The method of claim 25, wherein the request to reload is automatically generated by the status page at the client.

27. The method of claim 25, wherein the request to reload is manually generated at the client.

28. The method of claim 25, wherein the request to reload includes a set of data for retrying the database update.

29. The method of claim 24, further comprising storing a set of data for retrying the database update.

30. The method of claim 24, wherein retrying the database update includes rolling back the database update after a timeout period and then retrying the database update.

31. The method of claim 30, further comprising determining the timeout period in response to a timestamp contained in the status page.

32. A web transaction processing system, comprising:

database for holding a set of data associated with a web transaction;

server that obtains a form that includes a unique identifier for the web transaction and that initiates an update to the database in response to the form and that generates a log for the update such that the log is identified by the unique identifier, the server having a retry mechanism that obtains a request to reload a status page such that the request includes the unique identifier and that accesses the log in response to the request and retries the update if the log indicates a failure of the update such that the database update is performed at most once.

33. The web transaction processing system of claim 32, wherein the server obtains the form in a post command from a client and then provides the status page to the client such that the status page includes the unique identifier.

34. The web transaction processing system of claim 33, wherein the request to reload is automatically generated by the status page at the client.

35. The web transaction processing system of claim 33, wherein the request to reload is manually generated at the client.

36. The web transaction processing system of claim 33, wherein the request to reload includes the data for retrying the update.

37. The web transaction processing system of claim 32, wherein the retry mechanism rolls back the update after a timeout period and then retries the update.

38. The web transaction processing system of claim 37, wherein the retry mechanism determines the timeout period in response to a timestamp contained in the status page.

39. A web transaction system, comprising:

client that enters a set of data pertaining to a web transaction into a form such that the form includes a unique identifier for the web transaction;

server that obtains the form from the client and that initiates an update to a database in response to the form and that generates a log for the update such that the log is identified by the unique identifier, the server having a retry mechanism that obtains a request from the client to reload a status page such that the request includes the unique identifier and that accesses the log in response to the request and retries the update if the log indicates a failure of the update such that the database update is performed at most once.

40. The web transaction system of claim 39, wherein the server obtains the form in a post command from a client and then provides the status page to the client such that the status page includes the unique identifier.

41. The web transaction system of claim 40, wherein the request to reload is automatically generated by the status page at the client.

42. The web transaction system of claim 40, wherein the request to reload is manually generated at the client.

43. The web transaction system of claim 40, wherein the request to reload includes the data for retrying the update.

44. The web transaction system of claim 40, wherein the retry mechanism rolls back the update after a timeout period and then retries the update.

45. The web transaction system of claim 44, wherein the retry mechanism determines the timeout period in response to a timestamp contained in the status page.

IX. EVIDENCE APPENDIX

None.

X. RELATED PROCEEDINGS APPENDIX

None.